

SAS Techniques For Preparing Administrative Health Data For Research Use

Chel Hee Lee, MSc

Lisa M. Lix, PhD P.Stat.

School of Public Health

University of Saskatchewan

SAS User's Group – Saskatchewan

October 14, 2009

Administrative Health Data

- ✧ Administrative health data are collected for purposes of health system monitoring and physician remuneration, not for research purposes

- ✧ Advantages of using administrative health data for research
 - a) Population-based
 - b) Inexpensive to process and analyze compared to primary data
 - c) Can be used to address a variety of policy-relevant research questions

- ✧ Limitations of using administrative health data for research
 - a) Data are often of poor quality

Common Administrative Health Datasets

- ✧ Population registry
 - a) Contains health coverage, demographic, geographic, and socioeconomic information for health insurance registrants
 - b) An important tool for research because the data are used to identify study cohorts and covariates for statistical models

- ✧ Physician billing records
 - a) Contain fee-for-service claims submitted by physicians for remuneration

- ✧ Hospital separation abstracts
 - a) Contain information about events in acute care facilities

- ✧ Prescription drug file
 - a) Contain information about outpatient prescription drug dispensations

Cleaning Administrative Health Data

- ✧ Why is the data cleaning needed?
 - a) To enable comparisons across database
 - Identifiability and joinability
 - b) To evaluate quality improvement initiatives
 - Completeness, consistency, and validity
 - c) To ensure accuracy of statistical analyses

- ✧ Basic strategy
 - a) Identify characteristics of each variable that are consistent with high quality
 - Example: Values of SEX for a health insurance registrant should be consistent across study years
 - Example: Values of AGE for a health insurance registrant should increase incrementally across study years
 - b) Data retention is important – deleting records can affect generalizability of study results

An Example: Cleaning a Population Registry

- ✧ All health insurance registrants who have continuous health insurance coverage in each year of the period from 2004 to 2009 were eligible to be included in our study
- ✧ The registry contains a number of variables, including
 - a) ID - Anonymized personal health number
 - b) SEX
 - c) AGE
 - d) STUDY YEAR

ID	SEX	AGE	YEAR
102499	M	56.3	2007
102499	M	57.3	2005
102499	M	58.5	2006
102499	F	28.3	2004
303333	F	29.3	2005
303333	F	34.3	2007
303333	F	31.3	2007

Examples of Data Quality Problems

- ❖ Inconsistent value of SEX within the same registrant

ID	SEX	AGE	YEAR
102499	M	56.3	2007
102499	M	57.3	2005
102499	M	58.5	2006
102499	F	28.3	2004

ID	SEX	AGE	YEAR
303333	F	29.3	2005
303333	F	34.3	2007
303333	M	31.3	2007
303333	M	29.3	2005
303333	F	34.3	2007

Example of Data Quality Problems

- ✧ Anomalous values in the change in age across years

ID	SEX	AGE	YEAR
10098	M	34	2005
10098	M	35	2006
10098	M	40	2007
10098	M	37	2008
10098	M	38	2009

ID	SEX	AGE	YEAR
39993	F	66	2005
39993	F	67	2006
39993	F	64	2007
39993	F	69	2008
39993	F	70	2009

68

36

Using SAS for Data Cleaning

- ✧ We assume that you already know basic data steps
 - a) If-then-else statements
 - b) Techniques for merging datasets

- ✧ Use of temporary variables
 - a) Pattern recognition of sequential values across years
 - b) Decision making for identifying anomalous values

- ✧ SAS functions that are important for data cleaning
 - a) SEX - LAG, CATS, INDEX
 - b) AGE - LAG

LAG Function

- ✧ Returns values from a queue
- ✧ Syntax: LAG<n>(argument)
- ✧ Arguments
 - a) n : specifies the number of lagged values
 - b) argument: specifies a numeric or character constant or variable
- ✧ Example

ID	SEX	AGE	YEAR	LAG1	LAG2	LAG3
10098	M	34	2005	Missing	Missing	Missing
10098	M	35	2006	34	Missing	Missing
10098	M	40	2007	35	34	Missing
10098	M	37	2008	40	35	34
10098	M	38	2009	37	40	35

CATS Function

- ✧ Removes leading and trailing blanks, and returns a concatenated character string.
- ✧ Syntax: CATS(item-1<, ..., item-n>)
- ✧ Arguments
 - a) item: specifies a constant, variable, or expression.
- ✧ Example

```
data _null_;  
  string1 = "  Biostatistics  ";  
  string2 = "University of Saskatchewan";  
  string3 = "          School of Public Health";  
  sentence = CATS(string2, string3, string1);  
  put result $char;  
run;
```

Biostatistics, School of Public Health, University of Saskatchewan

INDEX Function

- ✧ Search a character expression for a string of characters, and return the position of the string's first character for the first occurrence of the string.

- ✧ Syntax: INDEX (source, excerpt)

- ✧ Argument
 - a) Source: specifies a character constant, variable, or expression to search
 - b) Excerpt: a character constant that specifies the string of characters to search for in source.

- ✧ Example
 - a) INDEX('11101', '0') = 4

How to clean inconsistent values of SEX

ID	SEX	LAG1	LAG2	1	2	Consistent?	3	4	Where?
235 (1)	F								
235 (2)	M	F							
235 (3)	M	M	F	T (1)	F(0)	N	10	2	$3 - 2 = 1$
ID	SEX	LAG1	LAG2	1	2	Consistent?	3	4	Where?
237 (1)	M								
237 (2)	F								
237 (3)	M	F	M	F (0)	T (1)	N	01	1	$3 - 1 = 2$

if last.id then do;

1. if sex eq lag1(sex) then decision1 = TRUE (1) or FALSE (0);
2. if sex eq lag2(sex) then decision2 = TRUE (1) or FALSE (0);
3. decision_sequence = cats(decision1, decision2);
4. position_zero = index(decision_sequence, '0');
5. the number of records within a subject – position_zero;

end;

How to clean inconsistent values of SEX

ID	SEX	LAG1	LAG2	1	2	Consistent?	3	4	Where?
235 (1)	M								
235 (2)	M	M							
235 (3)	F	M	M	F (0)	F(0)	N	00	1, 2	Last
ID	SEX	LAG1	LAG2	1	2	Consistent?	3	4	Where?
237 (1)	M								
237 (2)	M	M							
237 (3)	M	M	M	T (1)	T (1)	Y	11	0	None

The basic strategy is:

1. Find the pattern of values in each variable
2. Identify the inconsistent record(s)
3. Replace the inconsistent value with the correct value.

How to clean anomalous values of AGE

- Case 1. Anomalous value is found above the increasing pattern of the age sequence

ID	AGE	YEAR	Lag1 (age)	Lag1 (year)	1	2	3	4	5
235 (1)	34	2004	1		
235 (2)	35	2005	34	2001	1	1	1		
235 (3)	40	2006	35	2002	5	1	0	Predict age	
235 (4)	37	2007	40	2003	-3	1	0		
235 (5)	38	2008	37	2004	1	1	1		

- Age gap = age – lag1(age)
- Year gap = year – lag1(year)
- If age gap equal year gap then decision = TRUE(1) or FALSE(0)
- If (age gap less than 0) and (current decision equal to 0) and (previous decision equal to 0) then choose “PREVIOUS” record
- Correct age = current age – year gap;

Current data procedure

How to clean anomalous values of AGE

- Case 2. Anomalous value is found below the increasing pattern of the age sequence

ID	AGE	YEAR	Lag1 (age)	Lag1 (year)	1	2	3	4	5
235 (1)	66	2004	1		
235 (2)	67	2005	66	2001	1	1	1		
235 (3)	63	2006	67	2002	-4	1	0	Predict age	
235 (4)	69	2007	63	2003	6	1	0		
235 (5)	70	2008	69	2004	1	1	1		

- Age gap = age – lag1(age)
- Year gap = year – lag1(year)
- If age gap equal year gap then decision = TRUE(1) or FALSE(0)
- If (age gap less than 0) and (current decision equal to 0) and (previous decision equal to 1) then choose “CURRENT” record
- Correct age = current age + year gap;

Current data procedure

Thank you

Questions?